SystemX

Squid Configuration

- Accueil
- Linux
- Squid SquidGuard
- Configuration de base
  - *Réseau*
  - *Disque*
  - *Gestion des objets*
    - *Politique de gestion*
    - *Taille des objets*
- Contrôle d'accès
  - *Les ACL*
  - *Contrôle d'accès*
  - *URL non cachées*
- Autres paramètres
- Authentification externe
  - *Trouver de l'aide*
  - *PAM*
  - *LDAP*
    - *LDAP standard*
    - *Active Directory*
    - *Configuration commune*
- Configuration des clients
  - *Script de configuration*
  - *Découverte automatique*
  - *Configurations Linux*
    - *Variable d'environnement*
    - *apt-get aptitude*

Configuration de base

Le seul fichier de configuration se trouve dans **`/etc/squid3/squid.conf`**. C'est un peu monolythique car un fichier d'une centaine de lignes est courant mais au moins on a tous sous la main !

ATTENTION : cette configuration est destinée à la version 3. Concernant la version précédente (2.6) les différences sont mineures.

Réseau

Le premier paramètre **`http_port`** fixe le port d'écoute à 3128, les autres paramètres **`icp_port`** et **`htcp_port`** désacivent les services icp et htcp inutiles quand on a qu'un serveur.

```
http_port 3128
icp_port 0
htcp_port 0
visible_hostname squid
cache_mgr admin@systemx.fr
```

On pourra avantageusement utiliser les paramètres suivants pour améliorer la vitesse de résolution DNS :

- **`positive_dns_ttl 1 day`** : une adresse existante sera conservée en cache 1 jour.
- **`negative_dns_ttl 1 hour`** : une adresse inexistante sera conservée en cache 1 heure. Si l'adresse devait redevenir accessible dans moins d'une heure il faudrait rafraichir le cache pour pouvoir accéder à cette adresse.

Disque

je vous conseille de créer un *filesystem* dédié (environ 10Go) que l'on *montera* sur **`/proxy`**.

je vous conseille aussi d'utiliser le user *proxy* (créé par défaut lors de l'installation du paquet *squid3*) pour tous les fichiers utilisés par *squid* et lui affecter un *home directory* (dans notre cas **`/proxy`**) et un shell : **`/bin/bash`**.

Il faut indiquer où stocker les informations relatives à l'activité : **`access_log`**, **`store_log`**, **`cache_store_log`** et celles relatives au cache : **`cache_dir`**.

```
cache_effective_user proxy
cache_effective_group proxy
cache_dir ufs /proxy/squid/cache 50000 32 512
access_log /proxy/squid/log/access.log
cache_log /proxy/squid/log/cache.log
cache_store_log /proxy/squid/log/store.log
```

Les paramètres de **`cache_dir`** sont :

- **`ufs`** : format de stockage (ne pas modifier).
- **`/proxy/squid/cache`** : répertoire du cache.
- **`50000`** : taille du cache (par défaut 100 Mo) ici 5Go. Indiquer au maximum 80% de l'espace disque. Si vous stockez les logs sur le même *filesystem* descendez à 50%.
- **`32`** : nombre de répertoires de premier niveau (par défaut 16).
- **`512`** : nombre de répertoires de second niveau (par défaut 256). Dans notre cas on a 2^{14} répertoires.

Pour créer l'arborescence adéquate : **`squid3 -z`**.

Gestion des objets

Politique de gestion

Les paramètres suivants fixent l'utilisation et la taille des divers objets du cache.

```
cache_swap_low 90
cache_swap_high 95
```

```
ipcache_low 90
ipcache_high 95
```

Lorsque, respectivement, la taille des objets et des adresses IP stockées dans le cache atteignent, en %, la valeur *low* du cache les informations stockées commencent à être remplacées. Lorsque la valeur *high* est atteinte le remplacement devient plus agressif !

Taille des objets

Les valeurs pour les objets sont spécifiées en Kb et les valeurs pour les adresses en nombre d'entrées.

```
maximum_object_size 8192
minimum_object_size 0
maximum_object_size_in_memory 8
ipcache_size 4096
fqdncache_size 4096
```

Dans notre cas les objets de taille supérieure à 8 Mb ne seront pas stockées dans le cache, et tous les objets de taille infèrieure seront stockés puisque la valeur minimale est à 0. Les objets de taille infèrieure à 8 Kb pourront être gardés en mémoire.
Seront aussi stockées en cache 4096 adresses IP et 4096 noms qualifiés ce qui a pour but d'accélérer la résolution DNS.

Contrôle d'accès

Les ACLs

Il faut définir des règles nommées suivies d'un couple "clée valeur". Par défaut on pourra utiliser les suivantes :

```
acl manager proto cache_object
acl all src 0.0.0.0/0.0.0.0
acl our_networks src 127.0.0.1/32 10.0.0.0/8 172.16.0.0/12 192.168.0.0/16
acl our_servers src 192.168.0.100 192.168.0.110
acl our_destip dst 207.46.225.221 209.85.227.104
acl our_destdom dstdomain .eset.com .avast.com
acl our_bandst dstdomain .youtube.com .deezer.com .facbook.com .myspace.com
.msn.com .msn.fr
acl to_localhost dst 127.0.0.0/8
acl Safe_ports port 80
acl Safe_ports port 443
acl Safe_ports port 1025-65535
```

C’est assez explicite pour se passer de trop longs commentaires ...

- **`acl manager proto cache_object`** : le protocole *cache_object*
- **`acl all src 0.0.0.0/0.0.0.0`** : cette *acl* pose maintenant problème, le mot clé **`all`** est réservé. On peut la supprimer.
- **`acl our_networks src 127.0.0.1/32 10.0.0.0/8 172.16.0.0/12 192.168.0.0/16`** : toutes les adresses du réseau local.
- **`acl our_servers src 192.168.0.100 192.168.0.110`** : quelques serveurs qui bénéficieront d’un traitement spécial.
- **`acl our_destip dst 207.46.225.221 209.85.227.104`** : quelques destinations par adresses IP qui bénéficieront d’un traitement spécial (autorisation).
- **`acl our_destdom dstdomain eset.com avast.com`** : quelques destinations par noms de domaines (ici des antivirus) qui bénéficieront d’un traitement spécial (autorisation).
- **`acl our_bandst dstdomain .youtube.com ...`** : quelques autres destinations par noms de domaines qui consomment de la bande passante et qui bénéficieront d’un traitement spécial (interdiction).
- **`acl Safe_ports port ..`** : les ports 80, 443 et de 1025 à 65535. On notera une autre maniè d’écrire une liste.

Le contrôle d'accès

Une fois ces ACLs définies il faudra appliquer des autorisations ou des interdictions. **ATTENTION** : l'ordre a une importance capitale ! Enfin il vaudra mieux définir ce qui est autorisé et interdire tout le reste plutôt que l'inverse.

```
# le protocole cache_object (acl manager) ne pourra accèder que depuis l'adresse
127.0.0.1 (acl localhost) et pas depuis une autre
http_access allow manager localhost
http_access deny manager

# Interdit l'accès aux ports autres que 80, 21, 443 et de 1025 à 65535 (négation
```

```
(!) de acl Safe_ports)
http_access deny !Safe_ports

# Interdit l'accès vers 127.0.0.0/8 (acl to_localhost)
http_access deny to_localhost

# Interdit l'accès vers les domaines définies dans acl our_bandst
http_access deny our_bandst

# Autorise l'accès depuis les serveurs définis dans acl our_servers
http_access allow our_servers

# Autorise l'accès vers les adresses définies dans acl our_destip
http_access allow our_destip

# Autorise l'accès vers les domaines définis dans acl our_destdomain
http_access allow our_destdomain

# Autorise l'accès depuis mon réseau (acl our_networks)
http_access allow our_networks

# Interdit tout le reste (mot réservé all)
http_access deny all
```

Enfin on indiquera que Squid répondra à toutes les requêtes des clients : **`http_reply_access allow all`**.

Il est bon de remarquer que dans ce casi les sources *our_servers* et les destinations *our_dstip*, *our_dstdomain* seront toujours accessibles car si l'analyse des règles arrive jusqu'à ce niveau elle sera, dans tous les cas couronnée de succés lors de l'évaluation de la source *our_network*. Ces règles ne sont là que pour la suite, lorsque il y aura une authentification.

Les URLs à ne pas cacher

Lorsque l'on a affaire à du contenu dynamique, par exemple fabriqué à partir de scripts *CGI* auxquels on passe des paramètres identifiés par : *?*, il est inutle de cacher ce contenu car il n'est pas reutilisable. On placera donc **avant** les ACLs :

```
hierarchy_stoplist cgi-bin ?
acl QUERY urlpath_regex cgi-bin \?
cache deny QUERY
```

Quelques autres paramètres

- **`negative_ttl 30 minutes`** : si une page n'a pas été trouvée ou si une connexion a été refusée, une nouvelle tentative d'accés dans moins de 30 minutes donnera le même résultat même si la page est, entre temps, redevenue accessible.
- **`client_lifetime 12 hours`** : temps pendant lequel un navigateur peut rester connecté au cache.
- **`uri_whitespace deny`** : interdit les URLs contenant des espaces.
- **`logfile_rotate 5`** : la commande : **`squid3 -k rotate`** a le même effet que la commande **`logrotate`** sur les fichiers de logs (**`acces`**, **`cache`** et **`store`**).

Authentification externe

Trouver de l'aide

Les documentations sont lá : **`/usr/share/doc/squid/`** accessible par **`zcat`** sur les fichiers : **`README.auth_module.pam.gz`** et **`README.auth_module.ldap.gz`**. Toutefois les informations contenues sont plutot pauvres ...

Les commandes **`man squid3_pam_auth`** et **`man squid3_ldap_auth`** donnent plus d'informations.

PAM

C'est **`pam_auth`** qui permet l'authentification via *PAM*. Pour tester : lancer le programme. Il n'y a pas de prompt, tapez votre login et votre mot de passe séparés par un espace puis validez. Si ça ne répond pas *OK* vous avez un problème d'authentification qu'il faut absolument résoudre avant d'aller plus loin.

```
/usr/lib/squid3/pam_auth
mtemouil deepsecret
OK
^C
```

Vous vous êtes bien débrouillé !

La configuration de **squid.conf** maintenant :

```
auth_param basic program /usr/lib/squid3/pam_auth
```

LDAP

LDAP standard

On va tester si l'authentification via un annuaire LDAP standard fonctionne. On va faire une recherche avec la commande :

```
ldapsearch -x -LLL -P 3 -h mon_serveur_ldap -D
"uid=mon_login,ou=people,dc=systemx,dc=local" -w mon_mot_de_passe
```

Si vous avez le message *ldap_bind: invalid credential* inutile d'aller plus loin ... Sinon passez à la commande spécifique à *Squid* :

```
/usr/lib/squid3/squid_ldap_auth -v 3 -b "ou=people,dc=systemx,dc=local"
mon_serveur_ldap
```

Il n'y a pas de prompt, tapez votre login et votre mot de passe séparés par un espace puis validez. Si ça ne répond pas *OK* vous avez un problème d'authentification qu'il faut absolument résoudre avant d'aller plus loin.

Puisque tout va bien, on indique donc dans **squid.conf** :

```
auth_param basic program /usr/lib/squid3/squid_ldap_auth -v 3 -b
"ou=people,dc=systemx,dc=local" mon_serveur_ldap
```

Active Directory

Active Directory n'aime pas l'authentification anonyme, je vous conseille de créer un utilisateur qui servira à ça. Dans mon cas j'ai l'utilisateur *squid3* :

```
Nom complet                : squid3
nom d'ouverture de session : squid3
```

on teste :

```
ldapsearch -x -LLL -h mon_host_AD -D "cn=squid3,cn=Users,dc=systemx,dc=local" -b
"dc=systemx,dc=local" -w deep_secret
```

Comme pour le serveur LDAP standard si vous avez *ldap_bind: invalid credential* inutile d'aller plus loin ... Le mot de passe est incorrect ! Mais il y a d'autres raisons pour que l'authentification échoue. Une fois n'est pas coutume avec nos "amis" de Redmond le serveur *AD* renvoie des codes d'erreurs compréhensibles, ci apràs une liste (non exhaustive) des plus fréquents :

- **525** : user not found
- **52e** : invalid credentials
- **530** : not permitted to logon at this time
- **532** : password expired
- **533** : account disabled
- **701** : account expired
- **773** : user must reset password
- **775** : accound locked

Mais supposons que tout fonctionne, passons à l'authentification par *Squid* :

```
/usr/lib/squid3/squid_ldap_auth -h mon_host_AD -b "cn=users,dc=systemx,dc=local" -
D "cn=squid3,cn=users,dc=systemx,dc=local" -w deep_secret -f "(samaccountname=%s)"
```

Le paramètre : **-f "(samaccountname=%s)"** permet d'utiliser comme login le nom d'ouverture de session (**%s** est le login passé au programme d'authentification). Si vous voulez utiliser le nom complet il faudra remplacer **-f ...** par : **-u cn** (*cn = Canonical Name*).

Il n'y a pas de prompt, tapez votre login et votre mot de passe séparés par un espace puis validez. Si ça ne répond pas *OK* vous avez un problème d'authentification qu'il faut absolument résoudre avant d'aller plus loin.

Puisque tout va bien, on indique donc dans **squid.conf** :

```
auth_param basic program /usr/lib/squid3/squid_ldap_auth -h mon_host_AD -b
```

```
"cn=users,dc=systemx,dc=local" -D "cn=squid3,cn=users,dc=systemx,dc=local" -w
deep_secret -f "(samaccountname=%s)"
```

Configuration commune

```
# Le nombre d'occurences de squid_ldap_auth à lancer. Sur un serveur "chargé" il
en faudra plus
auth_param basic children 5

# Titre de la fenêtre d'authentification
auth_param basic realm Web-Identification

# Intervalle de temps pour la revalidation du mot de passe
auth_param basic credentialsttl 5 minutes

# La règle indispensable
acl ldap-auth proxy_auth REQUIRED
```

`proxy_auth` est un mot réservé et **`REQUIRED`** signifie *n'importe quel utilisateur valide*.

Il faut remplacer **`http_access allow our_networks`** par **`http_access allow ldap-auth`** et le tour est joué !

Attention, maintenant que ce ne sont plus les hosts de *our_networks* qui sont autorisés mais les utilisateurs authentifiés on peut vouloir autoriser quelques serveurs à ne pas s'authentifier. On va donc avoir les acls suivantes :

```
acl ...
...
acl ldap-auth proxy_auth REQUIRED
...
http_access ...
...
http_access deny our_bandst
http_access allow our_servers
http_access allow our_destip
http_access allow our_destdomain
...
http_access allow ldap-auth
http_access deny all
```

L'ordre de **`http_access ...`** est important (je sais je radote) ! Dans notre cas les destinations de *our_bandst* seront inaccessibles, et tout ce qui vient de *our_servers* ou est à destination de *our_destip* ou *our_dstdomain* se fera sans authentification.

Configuration des clients

Script de configuration

Votre navigateur préféré vous permet d'utiliser un serveur http pour charger un fichier de configuration automatiquei généralement nommé **`proxy.pac`** (l'extension *pac* pour Proxy Auto Configuration).

Un fichier *pac* est un fichier texte qui contient au moins une fonction JavaScript **`FindProxyForURL(url,host)`** qui retourne l'adresse et le port du proxy. Au minimum on doit avoir:

```
function FindProxyForURL(url, host) { return "PROXY proxy.fsystemx.local:3128;
DIRECT"; }
```

Le navigateur charge le fichier **`proxy.pac`**, exécute la fonction **`FindProxyForURL(url,host)`** en lui passant l'url et le host demandés (dans notre exemple ce ne sera pas utilisé).La fonction retournera la méthode de connexion, le navigateur contactera alors le proxy nommé : *proxy.systemx.local* sur le port 3128. Si le serveur proxy ne répond pas le navigateur sortira directement.

Voici un exemple plus complet qui doit marcher :

```
function FindProxyForURL(url, host)
{

    // Tout ce qui est à destination de company.com ne passe pas par le proxy
    if (dnsDomainIs(host,".company.com")) { return "DIRECT"; }

    if  (   isPlainHostName(host) ||
            dnsDomainIs(host,".systemx.local") ||
            isInNet(host, "10.0.0.0", "255.0.0.0") ||
            isInNet(host, "172.16.0.0", "255.240.0.0") ||
```

```
            isInNet(host, "192.168.0.0", "255.255.0.0")
        )
    {
        return "DIRECT";
    }
    else
    {
        if  (   (url.substring(0, 5) == "http:") ||
                (url.substring(0, 6) == "https:") ||
                (url.substring(0, 4) == "ftp:")
            )
        {
            return "PROXY squid:3128; proxy.free.fr; DIRECT";
        }
        else
        {
            return "DIRECT";
        }
    }
}
```

Si l'une des conditions suivantes est remplie :

- le nom du host contacté n'est pas pleinement qualifié (sans nom de domaine par exemple : *www*)
- le nom du domaine contacté est *systemx.local*
- le host contacté est dans un réseau conforme à la *RFC 1918*

alors le navigateur ne passera pas par le proxy.
Sinon, si aucune des conditions précédentes sont satisfaites, pour les protocole *http* et *ftp*, le navigateur passera par le proxy nommé *squid* à défaut par *proxy.free.fr* et enfin se connectera directement si aucun des proxys n'est disponible.
À défaut de la réalisation des conditions précédentes le navigateur se connectera directement.

Ce lien vous sera d'une grande utilité. Un peu moins austère, plus accessible et moins complet celui de Wikipedia : ici.

Découverte automatique

Le protocale **WPAD** (**W**eb **P**roxy **A**utomatic **D**iscovery) comme son nom l"indique permet de découvrir automatiquement la configuration vue ci dessus. Il suffit de créer dans votre *DNS* un alias du proxy nommé *wpad* et de créer un alias nommé **`wpad.dat`** du fichier **`proxy.pac`**.

Lorsque votre client de navigation démarre, s"il est configuré pour détecter automatiquement la configuration du proxy, il va exécuter : **`http://wpad.mondomaine.local/wpad.dat`**

Cas particulier Window 2008 : par défaut le serveur DNS de Microsoft bloque la résolution du nom "wpad" ... On se demande bien pourquoi ? Il semble que ce soit pour des raisons de sécurité mais vu la passoire qu'est déjà ce système, un peu plus un peu moins ... Vous pouvez changer ce comportement en mode commande :
Vérifier :
`dnscmd /info /globalqueryblocklist`
Supprimer :
`dnscmd /config /enableglobalqueryblocklist 0`
C'est un peu brutal comme méthode, si vous voulez faire dans la finesse regardez l'aide de **`dnscmd`**.

Configurations Linux

En ligne de commande on peut renseigner les applications sur la manière d'accéder à internet, nous allons voir comment.

Variable d'environnement

Il suffit d'affecter la variable :

```
export http_proxy=http://login:motdepasse@nom_du_serveur_proxy:3128
```

On préfèrera **`export`** à **`set`** car la variable sera héritée par les shells fils.

Tester :

```
w3m www.gogole.fr
```

Miraculeux non ?

apt-get, aptitude

Pour les utilitaires **`apt-get`** et **`aptitude`** il faut renseigner un des fichiers de l'arborescence **`/etc/apt/apt.conf.d`** soit créer le sien. On devra y trouver :

```
Acquire::http::Proxy "http://login:motdepasse@nom_du_serveur_proxy:3128";
```

En suivant la syntaxe ISC (fichier genre *named.conf* de *bind*) on peut l'écrire :

```
ACQUIRE {
        http {
             Proxy "http://login:motdepasse@nom_du_serveur_proxy:3128";
             }
        }
```

Tester :

```
aptitude install nom_du_paquet
```

Si vous ne savez pas quoi installer, essayez *htop*.

Dernière modification : le 04/04/2014

Site édité et hébergé par nous même sous licence Creative Common Contact : infos@systemx.fr
- SYSTEMX - SCM au capital de 750 € - SIRET 410 740 823 00035 Toulouse (31) -
- TVA intracommunautaire : FR53410740823 -